




jQuery事件处理

管理科学与工程学科
耿方方



主要内容

- 事件机制
- 页面载入事件
- 绑定事件
- 切换事件
- 移除事件
- 其他事件
- 事件应用

事件机制

- 当用户浏览页面时，浏览器会对页面代码进行解释或编译，这个过程实质上是通过事件来驱动的，即页面在加载时，执行一个load事件，在这个事件中实现浏览器编译页面代码的过程。事件无论在页面元素本身还是在元素与人机交互中，都占有十分重要的地位。
- 众所周知，页面在加载时，会触发Load事件，当用户单击某个按钮时，触发该按钮的Click事件，通过种种事件实现各项功能或执行某项操作。事件在元素对象与功能代码中起着重要的桥梁作用。
- 事件在触发后分为两个阶段，一个是捕获，另一个则是冒泡。大多数浏览器都不支持捕获阶段，jQuery也不支持。因此，在事件触发后往往执行冒泡过程。

事件机制

■ 案例1:

- `<script src="jQuery/jquery-3.2.1.js" type="text/javascript"></script>`

- `<script>`

- `$(function() {`

- `var intI=0;`

- `$("body, div, #btnShow").click(function() {`

- `intI++;`

- `$(".clsShow").show().html("您好, 欢迎来到jQuery世界!")`

- `.append("<div>执行次数" + intI + "</div>")`

- `});`

- `});`

- `</script>`

```
</head>
```

```
<body>
```

```
<div>
```

```
    <input id="btnShow" type="button" value="点击" class="btn">
```

```
</div>
```

```
<div class="clsShow"></div>
```

```
</body>
```

```
</html>
```

事件机制

- 在实际过程中，我们不需要事件的冒泡现象发生，可以通过 `stopPropagation()` 方法来实现，该方法可以阻止冒泡过程的发生。

```
■ <script>
■     $(function() {
■         var intI=0;
■         $("body,div,#btnShow").click(function() {
■             intI++;
■             $(".clsShow").show().html("您好,欢迎来到
jQuery世界!")
■             .append("<div>执行次数
<b>"+intI+"<b></div>");
■             event.stopPropagation(); });});
■ </script>
```

- 除了使用 `stopPropagation()` 方法外，还可以通过 `return false` 实现停止事件的冒泡。

页面载入事件

- 页面载入事件
- jQuery的页面载入事件ready () 方法类似于传统的JavaScript的onLoad ()方法。只是在执行时间上有区别。
- onLoad ()方法必须是页面中的全部元素完全加载到浏览器后才触发。如果用ready ()方法加载页面，则只要页面的DOM模型加载完毕，就会触发ready ()方法。
- ready ()的工作原理：在jQuery脚本加载到页面时，会设置一个isReady的标记，用于监听页面加载的进度，当然遇到执行ready ()方法时，通过查看isReady值是否被设置，如果未被设置，那么就说明页面未加载完成，在此情况下，将未完成的部分用一个数组缓存起来，当全部加载完成后，再将未完成的部分通过缓存一一执行。

页面载入事件

- 页面载入事件
- 在前面的章节中我们简单的介绍了jQuery中的页面载入事件ready()方法，除了简化的\$(function() {}))方法外，ready()方法还有以下几种不同的写法，但它们执行的效果是一样的。
- \$(document).ready(function() { //代码部分 });
- **\$(function() { //代码部分 });**
- jQuery(document).ready(function() { //代码部分 });
- jQuery(function() { //代码部分 });

绑定事件

- 在进行事件的绑定时，我们在前面使用了`.click(function() {})`绑定按钮的点击事件，除了这种写法之外，在jQuery中我们还可以使用`bind()`方法进行事件的绑定，`bind()`功能是为每个选择元素的事件绑定处理函数，其语法结构如下所示：
 - `bind(type, [data], fn);`
 - `type`为一个或多个类型的字符串，如“`click`”或“`change`”，也可以自定义类型：可以被参数`type`调用的类型包括`blur`、`focus`、`resize`、`scroll`、`unload`、`click`、`dblclick`、`mousedown`、`mouseup`、`mousemove`、`mouseover`、`mouseout`、`mouseenter`、`mouseleave`、`change`、`select`、`submit`、`keydown`、`keypress`、`keyup`、`error`。
 - `data`是作为`event.data`属性值传递给事件对象的额外数据对象。
 - `fn`是绑定到每个选择元素的事件中的处理函数。

绑定事件

- 如果要在一个元素中绑定多个事件，可以将事件用空格隔开，例如：

- ```
<script>
```
- ```
    $(function() {
```
- ```
 $("#btnBind").bind("click mouseout", function() {
```
- ```
            $(this).attr("disabled", "disabled");
```
- ```
 });
```
- ```
    });
```
- ```
</script>
```

# 绑定事件

- 案例2

- `<script src="jQuery/jquery-3.2.1.js" type="text/javascript"></script>`
- `<script>`
- `$(function() {`
- `$("#btnBind").bind("click", function() {`
- `$(this).attr("disabled", "disabled");`
- `});`
- `});`
- `</script>`
- `</head>`
- `<body>`
- `<input id="btnBind" type="button" value="Button" class="btn">`
- `</body>`

# 绑定事件

- 案例3

- `<script>`

- `$(function() {`
- `$(".txt").bind({focus:function() {`
- `$("#divTip").show().html("执行的是focus事件");`
- `},`
- `change:function() {`
- `$("#divTip").show().html("执行的是change事件");`
- `}`
- `});`
- `});`

- `</script>`

- `</head>`

- `<body>`

- `<div>姓名: <input type="text" class="txt"></div>`

- `<div id="divTip" class="clsTip"></div>`

- `</body>`

# 切换事件

- 在jQuery中，有两个方法用于事件的切换，一个是hover()，另一个是方法toggle()。所谓的切换事件，就是有两个以上的事件绑定于一个元素，在元素的行为动作间进行切换。
- 调用jQuery中的hover()方法可以使元素在鼠标悬停与鼠标移出的事件中进行切换，该方法在实现运用中，也可以通过jQuery中的事件mouseenter与mouseleave进行替换。下面的代码是等价的：

- `$( "a" ).hover(function() {`
- `//执行代码1}, function() {`
- `//执行代码2`
- `})`
- `$( "a" ).mouseenter(function() {`
- `//执行代码1`
- `})`
- `$( "a" ).mouseleave(function() {`
- `//执行代码2`
- `})`

hover()功能是在当鼠标移动到所选元素上面时，执行指定的第一个函数；当鼠标移出这个元素时，执行指定的第二个函数，其语法如下：

hover(over,out)

# 切换事件

- 案例4

- `<style>`

- `.clsFrame{border: 1px solid #666;width: 220px;}`

- `.clsTitle{background-color: #eee;padding: 5px;font-weight: bold;}`

- `.clsContent{padding: 5px;display: none;}`

- `</style>`

- `<script>`

- `$(function() {`

- `$(".clsTitle").hover(function() {`

- `$(".clsContent").show();},`

- `function() {`

- `$(".clsContent").hide();`

- `});`

- `});`

- `</script>`

```
<div class="clsFrame">
```

```
 <div class="clsTitle">jQuery简介</div>
```

```
 <div class="clsContent">Query是一个快速、
简洁的JavaScript框架，是继Prototype之后</div>
```

```
</div>
```

# 切换事件

- `toggle()` 方法，可以依次调用N个指定的函数，直到最后一个函数，然后重复对这些函数轮番调用。该方法在调用函数时并非随机或指定调用，而是通过函数设置的前后顺序进行调用，其调用的语法格式如下：
- `toggle (fn, fn2, [fn3, fn4, ...])`

# 切换事件

- 案例5
- `$(function() {`
- `$("img").toggle(function() {`
- `$("img").attr("src", "Images/img01. jpg");`
- `$("img").attr("title", this. src);`
- `}, function() {`
- `$("img").attr("src", "Images/img02. jpg");`
- `$("img").attr("title", this. src);`
- `},`
- `function() {`
- `$("img").attr("src", "Images/img03. jpg");`
- `});`
- `});`
- `</script>`

# 移除事件

- 在DOM对象的实际操作中，既然存在用于绑定事件的bind方法，那么相应的也存在用于移除绑定事件的方法，在jQuery中，可以通过unbind()方法移除绑定的所有事件或指定某一个事件，其语法结构如下所示：
- `unbind([type], [fn])`



# 移除事件

## ■ 案例6

- `<script>`
- `$(function() {`
- `$("p").click(function() {`
- `$(this).hide();`
- `});`
- `$("button").click(function() {`
- `$("p").unbind();`
- `});`
- `});`
- `</script>`
- `</head>`
- `<body>`
- `<p>这是一个段落。</p>`
- `<p>这是另一个段落。</p>`
- `<p>点击任何段落可以令其消失。包括本段落。</p>`
- `<button>删除 p 元素的事件处理器</button>`
- `</body>`

## 其他事件

- 除了以上的几种事件外，在jQuery中还有很多的事件处理方法，我们这里主要介绍一下最为实用的两种处理事件的方法：`one()`和`trigger()`。
- `one()`方法的功能是为所选的元素绑定一个仅触发一次的处理函数，其语法结构如下所示：
  - `one(type, [data], fn)`
- `trigger()`方法的功能是在所选择的元素上触发指定类型的事件，其语法结构如下所示：
  - `trigger(type, [data])`

# 其他事件

- 案例7:

```
<script>
```

```
 $(function() {
 function btn_Click() {
 this.value="65962530";
 }
 $("input").one("click", btn_Click);
 })
```

```
</script>
```

```
</head>
```

```
<body>
```

```
 <input id="Button1" type="button" value="点击查看联系方式" class="btn">
```

```
</body>
```

# 其他事件

- 案例8:

```
<script>

 $(function() {

 var oTxt=$("#input");

 oTxt.trigger("select");

 oTxt.bind("btn_Click",function() {

 var txt=$(this).val();

 $("#divTip").html(txt);

 });

 oTxt.trigger("btn_Click");

 })

</script>

</head>

<body>

 姓名:<input id="Text1" type="text" class="txt" value="河南中医药大学">

 <div id="divTip" style="padding-top: 5px;"></div>

</body>
```

# 事件应用

## ■ 案例9：文本框的事件应用

```
<style>
 body{font-size: 13px;}
 .divInIt{width: 390px;height: 55px;line-
height: 55px; padding-left: 20px;}
 .txtInIt{border: 1px solid #666;padding:
3px;}
 .spnInIt{width: 179px; height: 40px;line-
height: 40px;
float: right;margin-top: 8px;padding-left:
10px;}

 .divBlur{background-color: #FEEEC2;}
 .txtBlur{border: 1px solid #666;padding:
3px;}

 .divFocu{background-color: #EDFFD5;}
 .spnSucc{margin-top: 20px;}
</style>
```

# 表单应用

```
else{
 if(!chkEmail(vtxt)){
 $(this).removeClass("txtInIt").addClass("txtBlur");
 $("#email").removeClass("divFocu").addClass("divBlur");
 $("#spnTip").addClass("spnBlur").html("邮箱格式不正确!");
 }
 else{
 $(this).removeClass("txtBlur").addClass("txtInIt");
 $("#email").removeClass("divFocu");
 $("#spnTip").removeClass("spnBlur")
.addClass("spnSucc").html("");
 }
 });
 function chkEmail(strEmail){
 if(!/^w+[@]{1}w+[.]w+/.
.test(strEmail)){
 return false;
 }
 else{
 return true; }
 }
});</script>
```

# 事件应用

## ■ 案例10：网页选项卡事件应用

```
<script>
 $(function(){
 $("#menu li").each(function(index){
 $(this).click(function(){
 $("#menu
li.tabFocus").removeClass("tabFocus");
 $(this).addClass("tabFocus");
 $("#content
li:eq("+index+")").show()
 .siblings().hide();
 });
 });
 });
</script>
```

# 表单应用

```
<ul id="menu">
 <li class="tabFocus">家居
 电器
 二手

<ul id="content">
 <li class="conFocus">我是家居的内容
 欢迎来到电器城
 二手市场，产品丰富多彩

```

# 想一想

- 1、网页选项卡的原理？
- 2、动手编写网页选项卡？